Trojan Horse In Tally Server (Attack1)

Taxonomy:

Configuration, Change Management, Account Management

Applicability:

Tally Server

Method:

Assuming a well motivated and financed person or group seeks to alter the outcome of an election. They would then determine the target environment as outlined in the resource section, below. Once key positions one and two (see "Attack Team Core Personnel", below) were identified and brought into the attack team, enough sensitive information would be collected and/or readily available to reach “critical mass” and design, develop and implement the technical mechanisms (e.g. determine a trojan horse attack method, the payload to be carried, the interaction points with existing program structures, etc.).

Trojan horse code can generally be placed into one of two classes:
1) Closed-loop, or self-contained code that requires no additional input to execute its task.
2) Open-loop code that creates a communication channel that can be used at a later point in time to accept additional input (explicit commands, scripts, etc.) in order to alter operating system or tally server application operations, or accept additional, 2nd or 3rd generation closed-loop trojan horse code. Open-loop is often referred to as a “backdoor”, typically created by members of the original programming team to allow system access at some future time when their authorized access has been suspended. [ed: Matt, do you have reference material for this section?]

An open loop Trojan could be placed in the original source code of the tally server by the vendor, masked in such a way that it would be unlikely to be detected. A subsequent 2nd or 3rd generation Trojan, developed by the attack team to target an upcoming election, would be placed on the tally server as part of the preparation cycle of the server for the election. The Trojan would be designed to only activate during an election (e.g. only after midnight on the first Tuesday of November), and not function when the system was placed into an a testing mode.

Once triggered, the Trojan Horse code could modify the tally server's voting database directly, or by intercepting the communication path between the tally source [e.g. the memory card] "read" and the database "write" actions, and alter the totals. It would also need to perform some housekeeping tasks on tally server, such as altering audit logs, file timestamps, transaction records, etc., to mask the evidence of its own activities.

An additional or alternate function of the tally server Trojan could be to re-write the memory cards as they are presented, to reflect the falsified data. Since some jurisdictions may consider these cards as the only official record, they would be favored over the Direct Recording Electronic (DRE) voting system's paper receipts or any DRE on-board audit records. Even more disruptive is the possibility that an election official could help “lose” the paper records.

Resource Requirements:

Based on pre-election voting registration, published polling data, census data, and statistical analysis, the attack team would target counties or precincts by selection for where the anticipated voting margins are razor-thin or dead-even, and where there is a sufficient population to change an election outcome. The targets are further refined for 1) low-budget elections offices, where the resources for system security, such as staff, tools, or procedures, are most likely to be thinned or omitted; 2) busy population districts, where it is presumed that the compromise would have a better chance of escaping detection; 3) districts where voting equipment is known, and may lead to a preference in attempts to suborn technical personnel (i.e., where vendor X equipment is in use, and the attack planning team has existing relationships or contacts within the organization of vendor X).

Attack Team Core Personnel

1) Vendor personnel. Someone with technical duties, such as programming responsibilities, that provide access to the system architecture, source code, and possibly direct physical access. If they had support duties with one or more targeted county or state elections organizations, even better, as that reduces their distance from the live system to be compromised. Physical access is not strictly necessary, as the activation of a Trojan horse would fall to the elections insider (#2, below) who would have the last possible access to the system before the election. A substitute for Vendor personnel might be someone at the ITA or someone who had access to a stolen machine to reverse engineer the technology.

2) Technical Elections official: This would be a mid-level county elections official, either an original conspirator or someone who can be motivated to participate in the attack by outright bribery, self-advancement, or extreme political ideology. This official would posess knowledge and/or provide oversight of some or all of the elections systems, including the operating processes on the tally server, the design of the balloting, etc.

3) The financier: Someone has to pay the bills necessary to gather the intelligence on the first two positions above, fund the bribery, and pay for research and development. This could be an individual or small collection of people dedicated to achieving the goals of the attack, able to manage the project and willing to fund the efforts.

Additional Resource Roles: These roles will be expanded as necessary to overcome identified obstacles (below)

4) Polling place elections judge: In order to co-opt the paper receipts that the individual DRE’s

generate (Obstacle 1, Paper records)

5) Professional-caliber software developer: a “hired gun” capable of 1) writing Trojan code that was original, and not just a re-use of an existing exploit; 2) understanding the programming approach of the tally server application processes, and isolating vulnerabilities; 3) designing a coordinated attack that could dynamically react to a range of tallied data conditions and alter them in a way that creates a plausible false record. The more resources that are invested in technical expertise, the better crafted the attack would be, and thus the more likely to succeed. (Obstacle 2, Complexity)

Potential Gain:

To successfully alter the tabulated vote from some or all DRE’s in a county. A Trojan Horse attack would not have to explicitly reverse the recorded outcome of an election (e.g., create a republican victor in a predominantly democratic district) to be successful. It may be sufficient to simply alter a few undervotes, or reduce the margin of victory by a few votes. This would also be useful in voting events where a simple majority was not the determinate state, but where a ratio of votes (e.g., electing candidates across multiple possible positions, or as the basis for determining electoral representation) would be of enough interest to motivate the attackers.

Likelihood of Detection:

The Trojan Horse code could be detected in several places: by the vendor, by the test lab, or by an election official noticing anomalous results during a test or in a real election. A skilled programmer, however, will generally be able to hide a significant amount of dangerous code without being detected in testing. If hidden in the subroutine code of supporting Commercial Off-The-Shelf (COTS) libraries or software objects, the Trojan would not be discoverable by the test lab, as they are not expected to have access to the relevent COTS software. (See countermeasures.)

Obstacles to attack:

1. Paper Records. Each DRE generates a paper receipt with the tallies of the votes recorded. These paper records are forwarded to the tallying place as part of the official record, and should be cross-checked with the electronic tally figures.

Counteractant: See Additional Roles, #4, above.

2. Subject Matter Complexity. Elections processes are too specialized to be easily understood by an outsider.

Counteractant: See Additional Roles, #5, above.

3. Ballot Assembly Timeline. The dynamic nature of the ballot database might, at times, leave too small a window of opportunity for such an attack to be mounted.

Counteractant: See Additional Roles, #5, above.

4. Security Process. The tally server is an important piece of hardware, kept in a secure location at all times. All physical access is controlled by authorized personnel, and all logical access is carefully controlled and monitored. (Many counties are not organized to protect a tally server or detect intrusions against their servers.)

Counteractant: An effective, around-the-clock security process is 1) expensive, 2) only as effective as its weakest link, and 3) unlikely to be reviewed. Election cycles are awkward in that they do not fall frequently enough to make good security practice a habit, or to even assume that the same people will participate in sequential events. Security awareness in employees must be continually retrained and refreshed. And when the voting system has been owned by the state/county for several years, certain contempt for process (e.g." we've never had any security problems here") is inevitable.

Also, an elections official (even though corrupted) is likely to have enough credibility to overwhelm any question of a security risk in a parochial setting. This problem is exacerbated when the typically small community of elections professionals is taken into consideration. A recent annual study released by IDG and PricewaterhouseCoopers, surveying information security professionals, reported that current employees accounted for 33 percent of all security related threats, up from 28 percent reported in the same period last year.

Most importantly, a security process is only effective when it is being actively monitored and enforced. Too frequently, this process is foregone in the interests of cost and expediency. Mandatory post-election audits should examine all aspects of system security and elections process, to verify that procedures were followed, and to determine the procedure effectiveness.

5. Automated Security. A tally server employs a wide range of automated security controls, to prevent physical and logical tampering, and provide detection of unauthorized changes.

Counteractant: A tally server is most often a PC-based system, with all the known vulnerabilities that implies. Also, voting systems do not require even the basic review of security controls that traditional information systems employ.

It is also axiomatic among security and hacking professionals that once an attacker has physical access to an automated information system, any security control can be defeated.

6. The Voting System Testing Laboratory (VSTL) certification testing process. The certification process to be conducted by the VSTL could discover any Trojans included.

Counteractant: Any Source Code review is extremely expensive, and is not 100% effective. Code review is typically a two to five person-year effort, over several months, at a cost anywhere from $300,000 to $600,000 [N.B.: We need a verifiable source for the quantified cost]

Even where source code is provided to the VSTL, Trojan code can be placed in the underlying operating system, support libraries, and other objects that are not included in the certification review. [N.B.: Need Matt Bishop reference on program review and securing code]

In cases where the source code is not made available to the VSTL, review of compiled tally server application code is extremely difficult, and even more expensive, and vastly less effective.

Countermeasures:

- Visible inspection of the DRE receipts at the polling place
- Wide dispersal of DRE receipt data among elections judges, observers, and general public at the polling place, with instructions on how to verify these figures against officially posted election results.
- Pre-election simulations with rigorous statistical analysis run on tally servers.
- Use of dual tally servers during elections, preferably at different sites, to compare results.
- Use of tally server system security tools and least-privilege enforced access policy
- Installation of active configuration management software to monitor the tally server software and determine if changes have been made.
- Disabling communication channels for software and other inputs to the tally server systems, Removing network adapters and all external connectivity from voting machines, except for allowed connection to the tabulation center over a cryptographically secured link.

Citations:

http://www.redherring.com/Article.aspx?a=13555&hed=Security+Threats+Rise+22%25+&sector= Industries&subsector=SecurityAndDefense

NIST Special Publication SP 800-30, Risk Management Guide for Information Technology Systems